DOCUMENT RESUME

ED 250 174 SE 045 150

AUTHOR Libeskind, Shlomo; And Others

TITLE Logo Activities in Elementary Geometry.
INSTITUTION Montana Univ., Missoula. Dept. of Mathematics.
SPONS AGENCY National Science Foundation, Washington, D.C.
PUB DATE 83
GRANT SER-8160728
NOTE 50p.
PUB TYPE Guides - Classroom Use - Materials (For Learner) (051)

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS Computer Oriented Programs; Computer Software; Elementary Secondary Education; *Geometric Concepts; *Instructional Materials; Learning Activities; *Mathematics Education; *Mathematics Instruction; *Microcomputers; Preservice Teacher Education; *Programing; Teaching Methods
IDENTIFIERS *LOGO Programing Language

ABSTRACT

These activities were designed for use at the University of Montana, where they were tested for four quarters in a mathematics for elementary teachers course on informal geometry. They are for use with Apple II-Plus computers with 64K memory or Apple IIe computers and MIT Logo. (Modifications are necessary if the activities are to be used with Apple Logo or with MIT Logo for the Commodore 64.) Specific directions are given to the student for beginning to work, as well as for such activities as: turtle commands; designing procedures; workspace management; initializing disks and saving files on a disk; work with circles; recursion, POLY, and other procedures; constructions; and transformational geometry ideas. Generally, part of each page guides students to learn a procedure or idea, and then additional tasks or problems are posed. Finally, a summary of commands is given. (MNS)

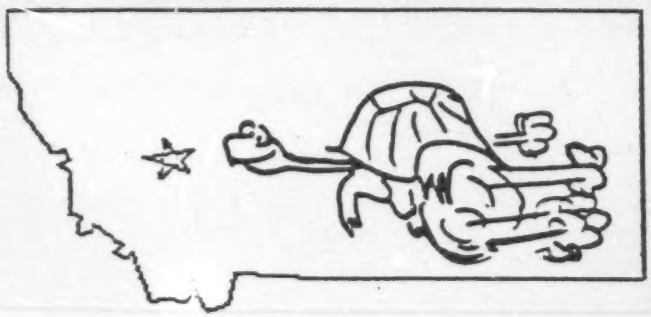
****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

X This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.



LOGO ACTIVITIES IN ELEMENTARY GEOMETRY

Shlomo Libeskind
Rick Billstein
Johnny W. Lott

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Johnny W. Lott

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

LOGO ACTIVITIES IN ELEMENTARY GEOMETRY

Shlomo Libeskind
Rick Billstein
Johnny W. Lott

Prepared Under Grant No. SER-8160728
Department of Mathematical Sciences
University of Montana
Missoula, MT 59812
© 1983

INTRODUCTION

These activities were designed for use at the University of Montana under NSF Grant Number SER-8160728. The activities were tested for four quarters in Math 133 Mathematics for Elementary Teachers (Informal Geometry). The activities are designed to be used with an Apple II-Plus with 64K memory or Apple II-e and MIT Logo. MIT Logo for the Apple is available through either Terrapin, Inc., or Krell Software Corporation. Modifications of the activities are necessary if they are to be used with Apple Logo or with MIT Logo for the Commodore 64.

1. TO BEGIN

(a) To load Logo, use the following steps.

1. Open disk drive door.
2. Insert Logo diskette into disk drive.
3. Close disk drive door.
4. Reach around to the back of the computer and turn it on.
5. Turn on the monitor.
6. When the disk drive has stopped whirring and the red light on the drive goes out, remove the Logo diskette and return it to its dust cover and put it in a safe place.
7. A welcome message should appear on the screen. This tells you the computer is ready for your input.

REMARK: When using the Apple II-e, make sure the CAP-LOCK key is depressed in order to make Logo work.

WARNING: Never place the diskette on the monitor since this may damage the data on the disk.

2. ARITHMETIC OPERATIONS IN NODRAW MODE

(a) The mode that Logo is in after the Logo system has been loaded is called the NODRAW mode. Arithmetic operations can be performed in this mode. Perform the computations in 1-6 by typing in each of the following and pressing RETURN after each arithmetic operation. If a typing error is made, press RETURN and retype the line.

1. 4*3+5
2. 4*(3+5)
3. 4*3+6/3
4. 4*3+6/3-2(3+2)
5. 4*(3+6)/(3-2)*3+2
6. 4*(3+6)/((3-2)*3+2)

REMARK: The RETURN key can be thought of as the "do it" key. Actions are not carried out until the RETURN key is pressed. If the screen becomes too cluttered, type CLEARTEXT to clear the screen.

(b) List the Logo symbol used to perform each of the following:

1. addition
2. subtraction
3. multiplication
4. division

(c) Does Logo appear to have an order of operations built into the machine, that is, are multiplications and divisions done before additions and subtractions? (Computations in parentheses are always done first.)

3. CORRECTING TYPING ERRORS

(a) To add 3+5, suppose you typed 3+7. To erase the 7, use the ESC key. The ESC key rubs out the character immediately to the left of the cursor (flashing rectangle) and moves the cursor one space to the left.

(b) To move the cursor left (right) without erasing any characters, use the arrow key <- (->).

(c) To add 3+5, suppose you typed 3+7. The flashing cursor can be moved to top of the 7 by pressing the left arrow key (<-). To delete a character over which the cursor is flashing, press the control key (CTRL) and while depressing that key, type D. This is called a CONTROL D or CTRL-D. (Other CTRL characters will be examined later.)

(d) To add 3+5, suppose you typed 2+5. You may move the cursor back to cover the 2 by holding the repeat key (REPT) down and depressing the left arrow key (<-). When the cursor is on top of 2, press CTRL-D. This erases the 2. Now type 3 and move the cursor to the desired position using the right arrow key (->). (Holding the REPT key down and pressing any other key causes the character generated by this key to be repeated.)

REMARK: The Apple II-e does not have a REPT key, but rather has an automatic repeat feature. If a key such as the key 2 is held down, the character is repeated until the key is released.

(e) Practice using the keys introduced above by completing the following.

1. Type 2+6+5; change this to 2+6+3.
2. Type 2+6+5; change this to 3+6+5.
3. Type 2+3*4; change this 2+3+4.
4. Type 2+5*6; change this to (2+5)*6.
5. Type the following sentence exactly as it appears:

   Someday I will on a komputr.

   Change this to read:

   Someday I will be able to type on a computer.

4. USING A PRINT COMMAND

(a) The Logo command PRINT may be used to print expressions on the monitor. PRINT is abbreviated by PR. Type each of the following and record your results. If an error message occurs, determine why. On the Apple II-plus, brackets can be obtained using SHIFT-M and SHIFT-N.

1. PRINT HELLO
2. PRINT "HELLO
3. PR HELLO"
4. PRINT "HELLO.JOE
5. PR "HELLO JOE
6. PRINT [HELLO]
7. PRINT [HELLO JOE]
8. (PRINT "HELLO "JOE)
9. Design a statement to make the computer print HI MOM.
10. Can you find another way to cause the computer to print HI MOM?

(b) Type the statements below into the computer. Record the results. How does the output differ when the PRINT (PR) command is used before the computation?

1. 3+5
2. PRINT 3+5
3. PR 3+5
4. PR "3+5

5. MEET THE TURTLE - THE DRAW MODE

(a) To enter the DRAW mode of Logo, type DRAW and press RETURN. The triangular object that appears is called the "turtle." To see how the turtle performs, type each of the following. Remember to press RETURN after each line. What does the turtle do in each case?

1.
```
FORWARD 50
RIGHT 90
FORWARD 50
RIGHT 90
FORWARD 50
RIGHT 90
FORWARD 50
```

2.
```
DRAW
FD 50
RT 90
FD 50
RT 90
FD 50
RT 90
FD 50
```

3. Type HOME

4. Type CLEARSCREEN (CS)

(b) Perform each of the following experiments.

1. Make the turtle draw a square with side of length 40.

2. Send the turtle home and clear the screen. Make the turtle draw a rectangle with sides of length 70 and 40.

3. With drawings on the screen, type DRAW. How does this result compare with:

   (a) Typing CLEARSCREEN (CS)

   (b) Typing HOME

   (c) Typing CLEARSCREEN followed by HOME

4. Experiment to see what happens if the space between the command and the input is omitted, e.g., FD50.

5. Experiment to see how the BACKWARD (BK) and LEFT (LT) commands work, e.g., BK 20 and LT 90.

6. Type FD -20 and BK 20. Is there a difference in the turtle's responses? Why?

7. Explore the dimensions of the screen in turtle steps.

6. OTHER TURTLE COMMANDS

Additional turtle commands are introduced in this activity.

(a) With the turtle at HOME, type FD 300. What happens?

(b) With the turtle at HOME, type NOWRAP and then type FD 300. Now what happens? To return to the wrapping behavior, type WRAP.

(c) Type RT 45 FD 1000. What happens? Why is a single line not drawn?

(d) With the turtle at home, type FD 140. The turtle appears to have disappeared. Type FULLSCREEN (CTRL-F) to see if you can find it. By typing SPLITSCREEN (CTRL-S), we return to the original mode.

(e) Another way to make the turtle disappear is by using the command HIDETURTLE (HT). For example, type the following:

```
FD 90
RT 60
HIDETURTLE
FD 50
RT 75
```

To make the turtle reappear, type SHOWTURTLE (ST). So far, the turtle has left a track everywhere it has gone on the screen. To make the turtle move without leaving a track, type PENUP (PU). Typing PENDOWN (PD) causes the turtle to again leave a track. Type a series of commands to obtain the following drawing. Make sure you hide the turtle at the end of the program.

7. DESIGNING PROCEDURES

(a) The EDIT mode in Logo is used to teach the computer new words. To enter the EDIT mode to draw a square, type TO SQUARE and press RETURN. The computer then enters the EDIT mode and will remember what you type if you follow the directions below. You can tell you are in the EDIT mode by the message that appears on the bottom of the screen. To complete the procedure, type the following:

```
FD 50
RT 90
FD 50
RT 90
FD 50
RT 90
FD 50
END
```

Now press CTRL-C. The procedure is then defined and a message to this effect is displayed.

(b) Try each of the following:

1. Type SQUARE.

2. Type SQUARE again. What happens" Do this two more times.

3. Use the HOME and CS commands or the DRAW command to clear the screen. Type SQUARE again. As long as the computer is not turned off, the computer will remember this procedure.

4. Type TO SQUARE again. The computer will enter the edit mode. Execute CTRL-N until the cursor is over the E of END. (To move the cursor up, we execute CTRL-P.) Execute CTRL-O to open a line. Now type RT 90 and then CTRL-C to define the new procedure.

   CTRL-N moves the cursor to the next line
   CTRL-O opens a line
   CTRL-P moves the cursor up to the previous line

5. Type SQUARE.

6. How does the new figure differ from the figure generated by the first SQUARE procedure?

7. A procedure can be aborted while still in the editor. To do this, a CTRL-G can be used. For example, enter the following.

```
TO JUNK
FD 50
RT 90
FD 50
END
```

Now instead of executing CTRL-C, use CTRL-G. Now type JUNK. Did the computer know how to JUNK? Enter the program again followed by CTRL-C. How does the computer know how to JUNK?

8. The command EDIT (ED) can be used instead of TO. Type EDIT RECTANGLE and write a procedure for drawing a rectangle of size 20 by 50 units.

9. Enter the EDIT mode and explore the commands CTRL-K, CTRL-A, and CTRL-E. What do these commands do?

8. USING THE REPEAT COMMAND

(a) A REPEAT command can be used in a Logo procedure. REPEAT takes two inputs - a number and a list of commands - and repeats the commands in the list the designated number of times.

1. Type TO SQUARE1
   REPEAT 4[FD 50 RT 90]
   END

2. Press CTRL-C.

3. Type SQUARE1.

4. Edit SQUARE1 by typing TO SQUARE1 or EDIT SQUARE1 to change the side to length 75.

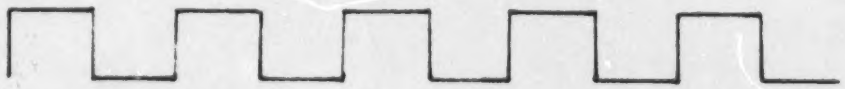
5. Run the edited SQUARE1.

(b) Use the REPEAT command to write procedures for drawing each of the following. Run your procedures to check that they draw the required figures.

1. A rectangle with sides of lengths 40 and 60.

2. An equilateral triangle with sides of length 50.

3. The following figure:



(c) More than one REPEAT command can be used in a procedure. For example, predict the outcome of the following procedure:

```
TO SPINSQUARE
REPEAT 10[REPEAT 4[FD 50 RT 90] RT 36]
END
```

(d) A REPEAT command can be used with a PRINT command. For example, type the following:

```
REPEAT 10 [PR "HELLO]
```

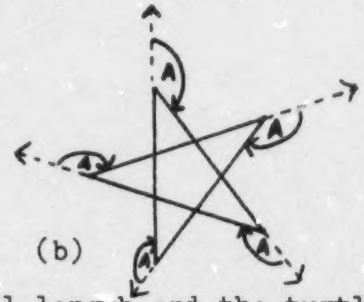
Procedures can be halted by using CTRL-G. For example, type
REPEAT 1000 [PR "HELLO]

and then press CTRL-G to halt the program.

9. CREATING A 5-POINTED STAR USING TRIAL AND ERROR

In this activity a five-pointed star which has sides of length 50 is constructed. This figure should appear as shown in Figure (a).

(a)



(b)

There are five sides of equal length and the turtle must turn through 5 angles of equal measure as shown in Figure (b). Thus, the program must look something like the one below where :A represents the measure of the angle.

```
TO STAR5 :A
REPEAT 5[FD 50 RT :A]
END
```

To determine the value of :A, consider Figure (b) above. Notice that the angle that the turtle turns each time must be greater than the angle it turned to form a square (90°) and less than a straight angle (180°). Because the angle appears greater than 120 as used in the equilateral triangle procedure, we might make an initial first guess of 140.

If the STAR5 procedure is run with an angle input of 140, the figure given below is obtained.

Because the figure is not closed, it appears that a larger angle should be used. The result of trying an angle of 150 is given below. From this figure observe that 150 is too large and the desired angle must be between 140 and 150.

(a) Experiment with other values in the STAR5 procedure until the correct value of the angle is determined. (You should hide the turtle in order to determine if your guesses are too large or too small.)

(b) Rewrite the STAR5 procedure to accept variable size inputs.

(c) Use the ideas developed above to write a procedure for drawing a 7-pointed star.

(d) Is it possible to draw a six-pointed star using this technique?

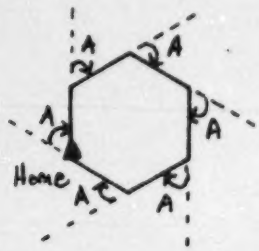
(e) Try to determine mathematically the correct angle for drawing various stars.

10. PROCEDURES WITH VARIABLES

In Logo it is possible to use a single procedure to draw figures of the same shape but different size. Consider a procedure which will draw squares of different size. It may have a variable input rather than a fixed number. To indicate a variable in Logo, we precede it with a colon. The following is a procedure to draw a variable size square. Type it into your computer and then execute (a) SQUARE 30; (b) SQUARE 50; (c) SQUARE 80.

```
TO SQUARE :SIDE
REPEAT 4[FD :SIDE RT 90]
END
```

To write a procedure which draws regular polygons other than squares, it is necessary to know the angle the turtle should turn through at each vertex. Consider the regular hexagon below.



Suppose the turtle starts at home, walks around the hexagon and comes back home and stops facing the same direction it started. If at each vertex the turtle turned by angle A, the turtle's total turning is $6 \cdot A$. Because the turtle walked around the hexagon and finished in its original position with its original heading, its total turning must be 360°. This is a special case of the Total Turtle Trip Theorem, which says that when the turtle walks around a closed path and ends at the same place and with the same heading as it started, its total turning is an integer multiple of 360°. Thus, in a hexagon. $6 \cdot A = 360^{\circ}$ and $A = 60^{\circ}$, so we have the following procedure for a variable size hexagon.

```
TO HEX :SIDE
REPEAT 6[FD :SIDE RT 60]
END
```

(a) What, if anything, is wrong with the following procedure? (Try to run the procedure for some input.) Correct the procedure to draw a square to the left of the turtle's initial position.

```
TO LSQUARE
REPEAT 4[FD :SIDE LT 90]
END
```

(b) Predict the outcome when the following is typed and the return key is pressed: SQUARE 50 SQUARE -50.

(c) Edit the SQUARE procedure so that the variable will be :S rather then :SIDE.

(d) Write a procedure for a variable size regular pentagon.

(e) Write a procedure for a variable size equilateral triangle.

(f) Use the triangle procedure in (e) to write a procedure which will draw the following figure made of 6 equilateral triangles.

(g) Write a single procedure for drawing a variable size regular n-gon (a polygon with n sides). If :N is the number of sides, your procedure should start with the line TO POLYGON :SIDE :N.

(h) Use the POLYGON procedure from (g) to draw a turtle type circle.

11. PROCEDURES CALLING PROCEDURES

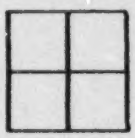
In Logo a procedure can be used within a procedure. For example, type the following into your computer:

```
TO SQUARE :SIDE
REPEAT 4[FD :SIDE RT 90]
END
TO SPINSQUARE :SIDE
REPEAT 10[SQUARE :SIDE RT 36]
END
```

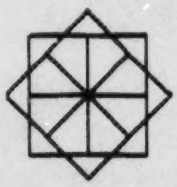
SPINSQUARE calls the previously defined SQUARE. Predict the kind of figure that will be drawn by SPINSQUARE 50 (do not type SPINSQUARE 50 before answering this question). Now type SPINSQUARE 50 and check your prediction.

(a) Edit the SPINSQUARE procedure and change the 10 in the REPEAT statement to 12. Run the edited procedure. Is a new figure obtained? Is the same result obtained if the 10 is changed to 20? to 9?

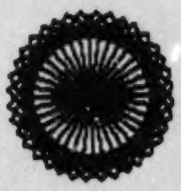
(b) Write a procedure similar to the SPINSQUARE procedure to draw each of the following.

(a) 

4 squares

(b) 

8 squares

(c) 

36 squares

(c) Write a procedure to draw an equilateral triangle of variable side length and then use this procedure to define a new procedure which will spin the triangle by 20° until the triangle returns to its original position. Run your procedure for three different side values.

12. WORKSPACE MANAGEMENT

(a) It is possible to define several procedures without leaving the editor by typing END after each procedure except the last one. For example, enter the editor by typing TO TRIANGLE :S and then continue as follows.

```
TO TRIANGLE :X
REPEAT 3[FD :X RT 120]
END
TO SQUARE :X
REPEAT 4[FD :X RT 90]
END
TO PENTAGON :X
REPEAT 5[FD :X RT 90]
END
```

When CTRL-C is executed, a message is displayed saying that all the procedures have been defined. CTRL-G can be used to abort the procedures.

(b) It is also possible to edit all the procedures in the workspace at once by typing EDIT ALL. If there are numerous procedures in the editor,the following commands will be helpful.

CTRL-F - scrolls the text forward one screenfull

CTRL-B - scrolls the text back one screenfull

CTRL-L - scrolls the text so that the line containing the cursor is approximately in the center of the screen

After changes have been made, CTRL-C is used to define all procedures.

Edit all the procedures in part (a) so that the turtle will be hidden after the figure has been drawn.

(c) The PRINTOUT (PO) command can be used to determine not only the listing of a procedure, but also to list all the procedures in the workspace. To obtain a listing of all the procedures in the workspace, type PRINTOUT TITLES (POTS).

To see a procedure listed without going into the editor, type PO along with the procedure name, for example, PO TRIANGLE.

The command PO ALL will display all the procedures in the workspace.

(d) To erase procedures from the workspace, use the command ERASE (ER). To eliminate single procedures such as TRIANGLE type ER TRIANGLE. To erase everything in the workspace type ER ALL. An alternative way to erase the entire workspace is to type GOODBYE.

(e) Experiment with the commands introduced above in order to see how they work.

13. INITIALIZING A DISK

In order to save procedures or pictures on a floppy disk, it must be initialized. The following instructions should be followed in order to initialize a disk.

(a) Place the Logo Utilities disk into the disk drive and turn on the power. Do not have Logo loaded into the computer. (Recall that the Logo Utilities disk is not the same as the Logo disk.)

(b) After the red light on the drive goes out, remove the Utilities disk from the drive and return it to its jacket.

(c) Insert the disk to be initialized into the disk drive.

(d) Type

INIT HELLO

and press RETURN. When the red light goes out, the disk has been initialized and can be used to store procedures and pictures.

(e) Remove the disk and label it with a felt tip pen. Never write on a disk with a sharp pen.

To make sure that a disk has been initialized, insert the disk into the drive and type CATALOG. The only file that should appear is the HELLO file and it should appear in the form shown below.

A 002 HELLO

REMARK: A disk can also be initialized by using the Apple System Master instead of the Logo Utilities disk. A minor drawback of this technique is that more space on the disk is used this way. It should be pointed out that students could just as well use their names instead of HELLO after the word INIT in Step (d). (Any initialized disk can be used to initialize a new disk.)

14. SAVING FILES ON A DISK

(a) Procedures can be saved on a disk under a file name. For example, type the procedures below into the computer.

```
TO TRIANGLE :X
REPEAT 3[FD :X RT 120]
END

TO SQUARE :X
REPEAT 4[FD :X RT 90]
END

TO PENTAGON :X
REPEAT 5[FD :X RT 72]
END
```

To save these three procedures in a file called POLYGONS, type the following.

```
SAVE "POLYGONS
```

The SAVE command not only saves these three procedures on the disk but also everything that is in the workspace at that time. Therefore, it is important that you type POTS before using the SAVE command in order to see exactly what procedures will be saved in the file. If there are unwanted procedures, they should be erased using the ERASE command before anything is saved on the disk.

When SAVE "POLYGONS is typed and the initialized disk is in the drive, the red light will turn on while the procedures are being saved on the disk. To check this, type CATALOG and the following appears.

```
A 002 HELLO
B 003 POLYGONS.LOGO
```

(b) Logo files are listed with .LOGO on the end of each file name. It is not necessary to add this when retrieving files. To see how to retrieve files, type GOODBYE to clear the workspace. If POTS is typed at this point, no procedures will be listed. To read a file into the computer's workspace, use the command READ as shown below.

```
READ "POLYGONS
```

When RETURN is pressed, the red light will come on and Logo will enter the procedures from the POLYGONS file into the workspace. Each procedure name will appear on the screen as it is entered. To erase the file use the ERASEFILE command along with the file name. For example, to erase the POLYGONS file, type

```
ERASEFILE "POLYGONS
```

(c) It is possible to save pictures on the disk using the SAVEPICT command. For example, draw a pentagon using PENTAGON 50. To save this picture on the disk, use the SAVEPICT command as follows.

```
SAVEPICT "PENTAGON
```

To see if the picture has been saved, type CATALOG. Pictures are saved with .PICT appended to their names. To display a picture which has been saved, use the READPICT command as follows.

```
READPICT "PENTAGON
```

It is handy to save pictures on the disk when it takes Logo a long time to draw them or when using a screen dump which sends pictures from the disk to the printer. Pictures take much storage room on the disk and thus you should limit the number of pictures stored on the disk. To erase the PENTAGON picture from the disk, type

```
ERASEPICT "PENTAGON
```

(d) Practice saving files and pictures on the disk and then erase them.

15. TURTLE-TYPE CIRCLES

The turtle can draw only figures made up of segments and hence it cannot draw a perfect circle. Since circles can be approximated by polygons, a regular polygon with a large number of sides will approximate a circle. A regular polygon on the screen resembles a circle and will be referred to as a turtle-type circle, or a circle. A typical approximation of a circle is done by a regular polygon of 360 sides. The procedure is given in (a).

(a) Type and run CIRCLE

```
TO CIRCLE
REPEAT 360[FD 1 RT 1]
END
```

(b) Write a procedure to draw a circle smaller than the one in (a).

(c) Write a procedure to draw a circle larger than the one in (a).

(d) Write a procedure to draw a variable size circle where the variable is the side, :S, of the polygon which approximates the circle. Call your procedure VCIRCLE (for Variable Circle).

(e) Type

```
TO CIRCLE1
REPEAT 90[FD 4 RT 4]
END
```

Which procedure, CIRCLE or CIRCLE1, runs faster? Why? Write a procedure to draw a circle which will be drawn faster than either CIRCLE or CIRCLE1.

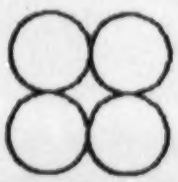
REMARK: Hiding the turtle causes a drawing to be done faster when a procedure is run.

(f) Write a procedure to draw a semicircle.

(g) Write a procedure to draw a variable size arc.

(h) Design a procedure to draw a circle to the left of home instead of the right of home.

(i) Write a procedure to draw the following.



16. A VARIABLE SIZE CIRCLE WITH RADIUS AS THE VARIABLE

(a) To draw a circle with radius 50 you may use the following procedure developed in 15(d).

```
TO VCIRCLE :S
REPEAT 360[FD :S RT 1]
END
```

Find what :S will produce a circle with radius 50. To do this write an equation for s and then solve for s.

We know that 360*s is the perimeter of the 360-gon with side s, which approximates the circle. Because $C = 2\pi r$, the circumference of the circle with radius 50 is $2\pi \cdot 50$. Hence, 360*s is approximately $2\pi \cdot 50$. That is,

$$360 * s \approx 2\pi * 50$$

$$s \approx \frac{\pi}{180} \cdot 50$$

$$s \approx 0.0174 \cdot 50$$

$$s \approx 0.87$$

Now run the above VCIRCLE for :S= 0.87.

(b) Use the method described in part (a) to draw a circle with radius 30.

(c) Next we write a procedure to draw a circle with radius :R. To accomplish this, :R can be substituted for 50 in the computation above; so we have : S ≈ 0.0174*:R. Run VCIRCLE .0174*50.

(d) A new procedure can be defined for drawing a circle with a given radius. If we call this procedure CIR with input :R, then the VCIRCLE procedure can be called to do the work of drawing the circle. The procedure CIR is given below.

```
TO CIR :R
VCIRCLE .0174*:R
END
```

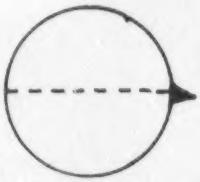
First run CIR 50 and then run the CIR procedure to draw several different size circles.

17. REVIEWING PROPERTIES OF CIRCLES

To determine if the turtle-type circle drawn by CIR 50 is of the desired radius, turn the turtle towards the center of the circle and let it walk along a diameter of 100 units. If the turtle ends up approximately on the turtle-type circle (at a diametrically opposite point) as shown in the figure below, then CIR 50 produced a good circle approximation.

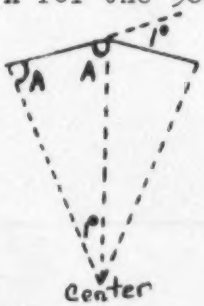
The position of the turtle after completing the circle.

The turtle facing the center.



The position of the turtle after its walk along a diameter.

After CIR 50 is run, how can you make the turtle turn towards the center? Recall that the turtle is actually walking along sides of a 360-gon and turning 1° at the end of each side. Two of the sides of the 360-gon are shown below. The vertices have been connected to the center of the circle. The turtle has to turn by angle A to head towards the center. Find A for the 360-gon.



(a) To make a figure which will be even more like a circle, it is necessary to draw a polygon with a greater number of sides. If the number of sides of the polygon is very large, what is the approximate measure of angle A?

(b) Edit the CIR procedure to make the circle and to make the turtle end up on a diametrically opposite point of the circle from its starting point by traversing the diameter.

(c) Write a procedure for drawing a circle with radius :R inscribed in a square as shown in the figure below.

(d) Read the ARCS procedures from the Logo Utilities disk and determine the function of each procedure. Save these procedures to your disk under the name ARCS.

18. INTRODUCING RECURSION

Logo allows one procedure to call another procedure; it also allows a procedure to call itself. A procedure that calls itself is called a recursive procedure. The following procedure demonstrates this phenomenon. To stop the procedure, type CTRL-G.

```
TO NONSENSE
PRINT [HELLO]
NONSENSE
END
```

(a) Predict what will happen and then run the procedure.

(b) Predict what kind of figure will be obtained by the following procedure and then check your guess by executing the procedure.

```
TO MOVE
FD 10 RT 36
MOVE
END
```

(c) Write a recursive procedure for drawing a circle.

(d) Write a recursive procedure for drawing a regular polygon with 20 sides of side length 20 units.

REMARK: When the recursive call is the last line of the procedure before the END statement, this recursion is referred to as tail-end recursion.

19. RECURSION WITH VARIABLES

(a) The following procedure draws hexagons of variable size. Type the procedure into the computer and run it for HEXAGON 40 and HEXAGON 80. (Remember CTRL-G will stop the procedure.)

```
TO HEXAGON :SIDE
FD :SIDE RT 60
HEXAGON :SIDE
END
```

(b) Write a recursive procedure for drawing a variable size pentagon and run it for two different values.

(c) The following procedure draws various figures, some of which are polygons. Run it for the following angles: 72, 60, 120, 144, 135, 108, 160.

```
TO FIGURE :ANGLE
FD 50 RT :ANGLE
FIGURE :ANGLE
END
```

(d) Predict for which angles the FIGURE program will draw polygons and for which FIGURE will not draw polygons. Test your prediction on the computer.

(e) Write a recursive procedure called POLY to draw figures as in part (c), but the size of the figure varies (your procedure must have two variables). Run your procedure at least five times, each time for different values of the variables.

20. USING HEADING TO STOP THE POLY PROCEDURE

The POLY procedure developed in Activity 19 may have been the following.

```
TO POLY :SIDE :ANGLE
FD :SIDE RT :ANGLE
POLY :SIDE :ANGLE
END
```

For any **integer** inputs POLY will run until we use CTRL-G. How can we stop POLY without using CTRL-G? One way is to use the primitive HEADING which outputs the turtle's heading at any given time. If the turtle starts with heading 0, then we should stop the procedure when heading is 0 again. Thus, a possible way to stop execution of the POLY procedure is by inserting the following line in our procedure.

```
IF HEADING = 0 STOP
```

(a) Predict the outcome if the line IF HEADING = 0 STOP is inserted in each of the following locations and check your predictions.

1. After the first line of POLY.

2. After the second line in POLY.

3. After the third line in POLY.

(b) Based on your answers to part (a), write a version of the POLY procedure which will stop. Try your procedure for POLY 40 72.

(c) Give the turtle a heading of 45 and now try POLY 40 72. Will this procedure stop? Why or why not?

21. OTHER WAYS TO STOP THE POLY PROCEDURE

In the previous activity we discovered that if the turtle's initial heading is not 0, the procedure may not stop. This activity deals with the problem of stopping POLY regardless of the turtle's initial heading. We consider two different approaches to this problem.

The first approach uses the Logo command MAKE. The statement MAKE "H HEADING gives the variable :H a value equal to whatever heading the turtle has at a given time. In general, the MAKE statement accepts two inputs: the first is the name and the second is the thing being named. The name in a MAKE statement must be preceded by a single set of quotation marks. Once a name is given in a MAKE statement, then when the name is used later in a procedure, it must be preceded by a colon.

(a) Predict the outcomes when the following are executed and then check your predictions on the computer.

1.
```
MAKE "B "BOAT
PRINT :B
PRINT "B
PRINT B
```

2.
```
TO NEXT :X
PR :X
MAKE "X :X+1
NEXT :X
END
```

3. Edit the NEXT procedure in (2) to print only the values 1-20.

(b) Consider the following procedure.

```
TO POLYSTOPA :SIDE :ANGLE
MAKE "H HEADING
FD :SIDE RT :ANGLE
IF HEADING = :H STOP
POLYSTOPA :SIDE :ANGLE
END
```

Predict what happens if the turtle is at home with heading 0 and the following is executed: RT 20 POLYSTOPA 50 144. Check your prediction on the computer. The procedure can be executed one step at a time using TRACE. Type DRAW, TRACE, and then execute RT 20 POLYSTOPA 50 144 again. To execute the trace feature, type any character other than CTRL-G or CTRL-Z. This will cause Logo to proceed to the next line in the procedure and wait. Typing any character again will cause Logo to execute the next line and so on. Observe that a new heading is obtained with each execution of the procedure. To quit the TRACE mode, type NOTRACE after the procedure is terminated.

POLYSTOPA does not stop the drawing because the MAKE statement changes the value of :H with each recursive call. If the MAKE statement is put in a separate procedure, START, outside POLYSTOPA, then a new procedure called POLYSTOP can be defined to stop the drawing. The START procedure, the edited POLYSTOPA procedure and the POLYSTOP are given below.

(c) Predict the outcome when RT 20 POLYSTOP 50 144 is executed.

```
TO START
MAKE "H HEADING
END

TO POLYSTOPA :SIDE :ANGLE
FD :SIDE RT :ANGLE
IF HEADING = :H STOP
POLYSTOPA :SIDE :ANGLE
END

TO POLYSTOP :SIDE :ANGLE
START
POLYSTOPA :SIDE :ANGLE
END
```

(d) Predict the outcome when the following is executed.

```
REPEAT 10[POLYSTOP 60 72 RT 36]
```

Check your prediction on the computer.

(e) Assume that the turtle's initial heading is 0. Write a **procedure called ROLL with integer inputs :SIDE, :ANGLE1, and :ANGLE2, which will "spin" the POLYSTOP figures by :ANGLE2. Incorporate a STOP statement in your procedure.**

(f) A different way to stop the POLY procedure regardless of the turtle's initial heading uses the concept of a counter and the Logo primitive REMAINDER. REMAINDER takes two positive integers as inputs and outputs the remainder when the first integer is divided by the second. For example, REMAINDER 20 6 outputs 2.

The POLY procedure should be stopped the first time the total turning is a multiple of 360, that is, when REMAINDER of total turning and 360 is 0. Thus, the following edited POLY procedure can be written.

```
TO POLY :S :A :N
FD :S RT :A
IF (REMAINDER :N*:A 360) = 0 STOP
POLY :S :A :N+1
END
```

Predict what happens when each of the following are executed and check your predictions on the computer.

1. RT 20 POLY 50 60 1
2. RT 40 POLY 50 144 1
3. RT 30 POLY 50 60 6
4. RT 40 POLY 50 144 0

22. VARIATIONS OF THE POLY PROCEDURE

The following POLY procedure can be modified in various ways to produce new figures.

```
TO POLY :SIDE :ANGLE
FD :SIDE RT :ANGLE
POLY :SIDE :ANGLE
END
```

For example, what happens when POLY calls itself again with different inputs? The following POLYSPI procedure is a result of calling POLYSPI with an input :SIDE+3 rather than :SIDE.

```
TO POLYSPI :SIDE :ANGLE
FD :SIDE RT :ANGLE
POLYSPI :SIDE+3 :ANGLE
END
```

Type the POLYSPI procedure and answer the following questions.

(a) Predict the outcome of POLYSPI 1 90 and check your prediction. Does POLYSPI 1 87 yield a similar result?

(b) Predict the outcome of POLYSPI 1 120 and check your answer. Does POLYSPI 1 118 yield a similar result?

(c) Edit POLYSPI to include a STOP statement which will make the procedure stop before a side longer than 100 units is drawn.

(d) A more general POLYSPI procedure has 3 inputs: :SIDE, :ANGLE, and :INC (for increment) and calls itself with :SIDE+:INC rather than :SIDE+3. Edit POLYSPI accordingly and run it several times.

(e) Another variation of the POLY procedure increments the angle rather than a side. Call the procedure INSPI and write it for variable inputs :SIDE :ANGLE :INC. Run the following.

1. INSPI 5 0 11
2. INSPI 5 0 5
3. INSPI 7 45 17
4. INSPI 3 45 30
5. INSPI 10 4 20
6. Your own inputs which will make a closed figure.

(f) Another way to modify the POLY procedure is to repeat the sequence FD :SIDE RT :ANGLE FD :SIDE RT :ANGLE*2 forever. Such a procedure called POLY1 follows.

```
TO POLY1 :SIDE :ANGLE
FD :SIDE RT :ANGLE FD :SIDE RT :ANGLE*2
POLY1 :SIDE :ANGLE
END
```

Run POLY1 for the following.

1. POLY1 15 144
2. POLY1 5 125

(g) Type in the following POLY2 procedure.

```
TO POLY2 :S :A
FD :S RT :A
POLY2 :S RT :A*2
END
```

Compare POLY2 15 144 and POLY2 5 125 with the runs in (f). Why are the outcomes different?

23. CONSTRUCTIONS

If possible, make the turtle construct each of the following:

(a) An angle with measure 70°

(b) The bisector of the angle in (a)

(c) A 50 "turtle-step" segment

(d) The perpendicular bisector of the segment in (c)

(e) A line segment through HOME

(f) A line segment parallel to the segment in (e) but 28 turtle steps above it

(g) A triangle with two sides of length 50 and 40 with measure of the included angle 83°

(h) A triangle with two angles having measure 60° and 83° and included side with length 50

(i) A triangle with two angles having measure 60° and 83° and not included side with length 50

(j) A triangle with two angles of measures 60° and 50°

★(k) A triangle with sides of length 50, 40, and 70
(A solution to this problem is developed in activity 33.)

24. STUDYING SYMMETRY

One of the most widely studied topics in elementary geometry is the notion of symmetry. Turtle graphics provides a mode for many investigations.

(a) Write a procedure to draw any figure, for example, a flower.

(b) Now write another procedure based upon your procedure in (a) in which you replace every RIGHT command with a LEFT command and every LEFT command with a RIGHT command.

(c) Run your procedure written in part (a). Without clearing the screen, run the procedure written in part (b). What do you observe?

(d) Repeat parts (a) - (c) using different procedures. Do you observe similar results?

(e) Rewrite the procedure in (a) to change all numerical inputs to their negatives. Clear the screen. Run the procedure in (a). Run the new procedure. What do you observe?

(f) In part (e) suppose you change only the inputs for FD and BK to their negatives. Do you have symmetry now?

(g) In part (e) suppose you change only the inputs for RIGHT and LEFT to their negatives. Do you have symmetry now?

(h) Another way to obtain symmetrical figures involves the use of <u>embedded recursion</u>, that is, recursion that is not tail-end recursion. An example of a procedure that contains such recursion is given below. Run this procedure for TOWER 30. Was the result expected?

```
TO TOWER :SIZE
IF :SIZE < 0 STOP
SQUARE :SIZE
FD :SIZE
TOWER :SIZE - 10
SQUARE :SIZE
FD :SIZE
END
```

```
TO SQUARE :SIZE
REPEAT 4[FD :SIZE RT 90]
END
```

(i) Write a procedure similar to the one in (h) that draws a series of symmetrical triangles.

(j) Try to write a procedure that draws a figure with a vertical line of symmetry using embedded recursion.

5. COORDINATIZING THE TURTLE

Turtle graphics contains many coordinate geometry capabilities. Many of these are investigated in this activity. The monitor screen can be thought of as containing an x- and y-axis with HOME as the origin with coordinates (0,0).

(a) SETX is a primitive that moves the turtle in a horizontal direction. For example, SETX 25 moves the turtle along a horizontal line to the point on the horizontal line where the x-coordinate is 25 and leaves the heading unchanged. Similarly, SETY moves the turtle in a vertical direction. Use SETX and SETY commands to write a procedure to draw a square with variable side.

(b) SETXY is a command that takes two numerical inputs and combines the SETX and SETY commands in a natural manner. For example, SETXY 25 25 moves the turtle to a point with coordinates (25,25). Write a procedure to draw the figure below using the SETXY, PU, and PD commands.

(c) Type SETXY -20 -30. What is the result? Why?

(d) SETXY can be used to draw a segment of a line through points with variable coordinates (:A,:B) and (:C,:D). Write a procedure to draw such segments.

(e) Write a procedure to draw a variable size rectangle using SETXY commands.

(f) Two other useful coordinate commands are XCOR and YCOR. These commands can be used to print the value of the x- and y-coordinates of the turtle no matter where the turtle is located. Draw any closed figure on the screen, put the pen up, hide the turtle, and then move the turtle around on the screen. Use XCOR and YCOR to find the coordinates of the turtle. Guess whether or not the turtle is inside the closed figure, and then show the turtle to determine whether or not you are correct. Try this little game several times.

(g) Two other coordinate commands are SETHEADING (SETH) and TOWARDS. SETHEADING takes a numerical input and turns the turtle to the indicated heading. For example, SETH 80 causes the turtle to turn to an 80 degree heading. TOWARDS is commonly used with SETHEADING. It takes two numerical inputs which are interpreted

as an x- and y-coordinate and turns the turtle towards the point with coordinates (x,y).

Recall that HEADING outputs the current heading of the turtle. Place the turtle anywhere on the screen and use HEADING to determine its heading. Estimate the coordinates of a point at which the turtle would have to be pointing to have a heading of 38 degrees. Check your guess using SETH TOWARDS and HEADING. Try this exercise several times and then try it with a heading different from 38 degrees.

26. WRITING NEW CIRCLE PROCEDURES

On this sheet, procedures involving coordinates are investigated. Two new commands are used: MAKE and SQRT. MAKE takes two items as input - the first becomes the name of the second. A set of quotation marks is used to designate the name to the computer. For example, MAKE "A 2+3 causes the variable :A to take the value of 2+3, or 5. SQRT takes one input and finds the square root of that input. The procedures CIRCLE and VCIRCLE developed previously are given below and are used to develop new procedures.

```
TO CIRCLE :R
VCIRCLE :R*.0174
END
```

```
TO VCIRCLE :S
REPEAT 360 [FD :S RT 1]
END
```

(a) To write a procedure for a circle whose center is at the origin (0,0) and passes through the point with coordinates (:X,:Y), the distance formula is used. Recall that the distance between any two points with coordinates (x1,y1) and (x2,y2) is

$d = \sqrt{(x1-x2)^2 + (y1-y2)^2}$. If the circle has radius :R, then :R must be the distance between the points (0,0) and (:X,:Y). The procedure may now partially be written as follows.

```
TO CIRCLE1 :X :Y
MAKE "R SQRT (:X*:X+:Y*:Y)
CIRCLE :R
END
```

Edit the CIRCLE1 procedure to mark the center of the circle and to actually make (0,0) the center of the circle.

(b) Write a procedure called CIRCLE2 to draw a circle, which takes four inputs :X1, :Y1, :X2, and :Y2, where (:X1,:Y1) are the coordinates of the center of the circle and (:X2,:Y2) are coordinates of a point on the circle. Use any of the previously developed procedures in writing the new procedure.

27. GENERATING SIMILAR FIGURES

Logo can be used to draw similar figures. Try the following activities.

(a) Write a program which will draw variable size triangles similar to the triangle with vertices at (0,0), (10,15), (20,0).

(b) Write a program which will draw a variable size rectangle similar to a rectangle with sides 20 and 40.

(c) Type in the following program for drawing a parallelogram with angle :A and sides :S1 and :S2, and then answer the questions below.

```
TO PARALLELOGRAM :A :S1 :S2
RT 90 - :A
FD :S1
RT :A
FD :S2
RT 180 - :A
FD :S1
RT :A
FD :S2
RT 90
END
```

1. Run each of the following:

   PARALLELOGRAM 70 20 40

   PARALLELOGRAM 70 30 33

   PARALLELOGRAM 70 50 40

   Are the three parallelograms similar? Why or why not?

2. Predict the outcome for PARALLELOGRAM 120 30 50 and check your prediction on the computer.

3. Use the PARALLELOGRAM program to draw three parallelograms similar to PARALLELOGRAM 50 20 30.

4. Write a procedure that will draw a variable size parallelogram similar to PARALLELOGRAM 50 20 30.

5. State an if and only if condition for two parallelograms to be similar.

(d) Draw a non-regular pentagon and two pentagons similar to it.

28. SLIDES - TRANSLATIONS

Slides are explored using an L-shaped figure drawn by the ELL procedure. Type the following into the computer and then try the activities. The variable :DIRECTION gives the heading in which the figure is to be moved. :DISTANCE is the variable representing the length of the slide.

```
TO SLIDE :DIRECTION :DISTANCE
FULLSCREEN
ELL
PU
SETH :DIRECTION
FD :DISTANCE
PD
SETH 0
ELL
SPLITSCREEN
END
```

```
TO ELL
FD 50
RT 90
FD 5
RT 90
FD 45
LT 90
FD 20
RT 90
FD 5
RT 90
FD 25
RT 90
HT
END
```

(a) Predict the results of each of the following procedures. Run each of them to see if you were correct.

1. SLIDE 45 40
2. SLIDE 180 60
3. SLIDE 300 80
4. SLIDE -45 30
5. SLIDE -270 (-50)
6. SLIDE 0 25

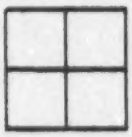
(b) If the ELL figure is drawn away from the "home" position, will the SLIDE procedure still work? Try it.

(c) Write a procedure to draw your own figure and modify the SLIDE procedure to perform slides using your figure.

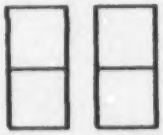
(d) Create patterns on the screen using the SLIDE procedure.

(e) Write a procedure to draw a square and then use the idea of a slide to create the following figures.

(a)



(b)



(f) Change the SLIDE procedure to draw the original L-shaped figure in the lower left corner of the screen, and have the turtle leave a dotted line showing the length of the slide.

(g) If a colored monitor is available, the primitive PENCOLOR (PC) may be used to draw the image in a different color from the original. Experiment with the PENCOLOR command with inputs 0-6 and then edit the SLIDE procedure to give an image in a second color.

REMARK: The background color can be changed using the primitive BACKGROUND (BG) and inputs 0-6. Experiment with pencolors and background colors.

29. TURNS - ROTATIONS

Type in the following programs and run the programs to answer the questions and try the activities. The variable :TURN.ANGLE represents the amount of the turn; :X and :Y are the coordinates of the center of the turn.

```
TO TURN :TURN.ANGLE :X :Y
FULLSCREEN
ELL
CENTER :X :Y
MAKE "D SQRT (:X* :X+ :Y* :Y)
SETH TOWARDS 0 0
MAKE "A HEADING
RT :TURN.ANGLE
FD :D
LT :A
PD
ELL
SPLITSCREEN
END
```

```
TO ELL
FD 50 RT 90
FD 5 RT 90
FD 45 LT 90
FD 20 RT 90
FD 5 RT 90
FD 25 RT 90
HT
END

TO CENTER :X :Y
PU SETXY :X :Y PD
FD 1 BK 1 PU
END
```

(a) Predict the results of each of the following procedures. Run each of them to see if you were correct.

1. TURN 45 (-20) (-40)
2. TURN 90 40 0
3. TURN 290 (-30) 20
4. TURN 90 25 50
5. TURN 180 0 0
6. TURN -60 30 20

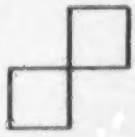
(b) Write a procedure to draw your own figure and then modify the TURN procedure to find its turn image. (Be careful where you leave the turtle.)

(c) Create a pattern on the screen using the TURN procedure.

(d) Create a drawing which makes use of both the SLIDE and the TURN procedures.

(e) Create the following figures by creating a procedure to draw a square and then using the ideas of the TURN procedure.

(a)



(b)

(f) Can you modify the TURN procedure so that it will draw a turn image when the original figure is not located at the "home" position?

30. LEARNING ANOTHER TURN PROCEDURE

A different procedure for performing a turn involves marking the center of the turn, moving forward away from the center of the turn the length of the radius of the "turn circle," drawing the picture that you wish to turn, and essentially moving along the circle with given radius and center and then drawing the image of the picture. A procedure to accomplish such a turn follows. The variables in the procedure are :TURN.ANGLE, the angle of the turn, :X and :Y, the coordinates of the center of the turn, and :R, the radius of the circle on which the picture is turning, that is, the distance from the turn center to a key point on the figure.

```
TO TURN2 :TURN.ANGLE :X :Y :R
CENTER :X :Y :R
ELL
PU
IF :TURN.ANGLE > 0 RT 90 RARC :R :TURN.ANGLE LT 90 ELSE LT 90
    LARC :R (-:TURN.ANGLE) RT 90
PD
ELL
END
```

```
TO CENTER :X :Y :R
PU
SETXY :X :Y
PD FD 1 BK 1 PU
FD :R
PD
END
```

```
TO ELL
FD 50 RT 90
FD 5 RT 90
FD 45 LT 90
FD 20 RT 90
FD 5 RT 90
FD 25 RT 90
HT
END
```

```
TO RARC :R :TURN.ANGLE
REPEAT :TURN.ANGLE [FD 3.14*:R/180 RT 1]
END
```

```
TO LARC :R :TURN.ANGLE
REPEAT :TURN.ANGLE [FD 3.14*:R/180 LT 1]
END
```

1. Predict the results of executing each of the following. Then use the computer to check your predictions.

(a) TURN2 90 0 0 50

(b) TURN2 180 0 0 50

(c) TURN2 0 0 0 50

(d) TURN2 -90 0 0 50

(e) TURN2 90 (-50) (-50) 50

(f) TURN2 30 80 90 40

2. Write a procedure to draw a picture other than the ELL. Edit the TURN2 procedure to turn your shape. (HINT: To avoid problems, make sure that the turtle returns to the original starting point when the drawing is finished.)

3. Edit the TURN2 procedure to show the turn angle as a dotted arc and to make the turn image a second color.

4. Compare the TURN2 procedure of this activity with the previous TURN procedure. Do they produce the same results?

5. What will happen if the following procedures are run one after another?

   (a) TURN2 60 0 0 30

   TURN2 60 0 0 -30

   (b) Execute the procedures in (a) as directed to determine if your predictions were correct.

6. (a) Start the turtle in the HOME position and run the following procedures one after the other.

   TURN2 70 0 0 30

   TURN2 50 0 0 30

   (b) Could the same result have been obtained by executing TURN2 50 0 0 30 followed by TURN2 70 0 0 30?

   (c) Can you design a single procedure to accomplish the same result as obtained in (a) and (b)?

31. FLIPS - REFLECTIONS

Flips are explored using an "L." Type the following program into the computer and then try the activities. The variable :DIRECTION sets the heading of the flip line; :DISTANCE is the distance from a key point of the L-shaped figure to the flip line. The procedure F.LINE draws the flip line. The procedure F.ELL draws a "reverse" image of ELL.

```
TO ELL
FD 50
RT 90
FD 5
RT 90
FD 45
LT 90
FD 20
RT 90
FD 5
RT 90
FD 25
RT 90
HT
END
```

```
TO F.LINE :DIRECTION
SETH :DIRECTION
FD 75
BK 150
FD 75
END
```

```
TO FLIP :DIRECTION :DISTANCE
F.LINE :DIRECTION
RT 90
PU
FD :DISTANCE
PD
SETH 0
ELL
PU
SETH (90+ :DIRECTION)
BK ( :DISTANCE*2)
LT (90- :DIRECTION)
PD
F.ELL
END
```

```
TO F.ELL
FD 50
LT 90
FD 5
LT 90
FD 45
RT 90
FD 20
LT 90
FD 5
LT 90
FD 25
LT 90
HT
END
```

(a) Predict the results of each of the following procedures. Run each of them to see if you were correct.

1. FLIP 0 60
2. FLIP 60 40
3. FLIP 40 60
4. FLIP 90 50
5. FLIP 180 50
6. FLIP 290 50

(b) Write a procedure to draw your own figure and modify the FLIP procedure to find its flip image. (Be careful where you leave the turtle.)

(c) Create a pattern on the screen using the FLIP procedure.

(d) Write a procedure which uses both a slide and a flip.

32. GLIDE REFLECTIONS

Glide reflections are explored using an L-shaped figure. :DIRECTION is the heading of the flip line. :DISTANCE is the distance from a key point of the figure to the flip line. :G.DISTANCE is the length of the slide. Enter the following programs into your workspace.

```
TO GLIDE :DIRECTION :DISTANCE :G.DISTANCE
FLIP :DIRECTION :DISTANCE
G.SLIDE :DIRECTION :G.DISTANCE
END
```

```
TO ELL
FD 50
RT 90
FD 5
RT 90
FD 45
LT 90
FD 20
RT 90
FD 5
RT 90
FD 25
RT 90
HT
END
```

```
TO G.SLIDE :DIRECTION :G.DISTANCE
FULLSCREEN
MAKE "TURN HEADING
PU
SETH :DIRECTION
FD :G.DISTANCE
PD
SETH :TURN
F.ELL
SPLITSCREEN
END
```

```
TO F.ELL
FD 50
LT 90
FD 5
LT 90
FD 45
LT 90
FD 20
LT 90
FD 5
LT 90
FD 25
LT 90
HT
END
```

(a) Predict the results of each of the following procedures. Run each of them to see if you were correct.

1. GLIDE 0 40 60
2. GLIDE 60 40 60
3. GLIDE 90 30 25
4. GLIDE 45 60 70
5. GLIDE (-45) (-40) (-60)

(b) Write a procedure to draw your own figure, and modify the GLIDE procedure to perform a glide reflection on it.

33. SSS TRIANGLE CONSTRUCTION

In an earlier activity, you were asked to write procedures for drawing specific triangles given various parts. In this activity, you are asked to write some general procedures and adapt others when given various parts of triangles. New commands used in these procedures are ANYOF, NOT, and OUTPUT (OP). ANYOF takes 2 or more inputs (default is 2) and outputs TRUE if any of the outputs are true. (What do you think ALLOF would do?) NOT outputs TRUE if its input is false and FALSE otherwise. OUTPUT takes one input and causes the current procedure to stop and output the result to the calling procedure.

Type the following procedures into your computer.

```
TO SSS :S1 :S2 :S3
IF NOT (ALLOF  ( :S1+ :S2> :S3)( :S1+ :S3> :S2)( :S2+ :S3> :S1)) PRINT
     [NO TRIANGLE IS POSSIBLE WITH THESE SIDES.] STOP
HT
FD :S1
PU
FD :S2
RT 90
CHECKSIDE
PD
HOME
FD :S1
SETXY :X :Y
HOME
END

TO CHECKSIDE
IF (ABS (:S3 - SQRT (XCOR*XCOR + YCOR*YCOR))<2) MAKE "X XCOR MAKE "Y
     YCOR STOP
FD :S2*.0174 RT 1
CHECKSIDE
END

TO ABS :VALUE
IF :VALUE<0 OUTPUT -:VALUE
OUTPUT :VALUE
END
```

(a) Run the procedures above for various lengths of sides for :S1, :S2, and :S3.

(b) What does the ABS procedure do? Where have you encountered this notion before?

(c) Why do we not just check to determine if :S3 is equal to SQRT (:X*:X+:Y*:Y)?

(d) The SSS procedure depends upon one vertex being at home. Edit the procedures above to draw a triangle with one of its vertices at the point with coordinates (:A,:B).

34. STILL MORE TRIANGLE CONSTRUCTIONS

Another set of procedures can be developed for drawing a triangle when two angles and an included side of the triangle are given. Such procedures are given below.

```
TO ASA :A1 :S :A2
IF NOT (:A1 + :A2 < 180) STOP
BK :S
RT :A1
CHECKHEAD
HOME
END

TO CHECKHEAD
SETH TOWARDS 0 0
IF ABS (HEADING - (360- :A2)) < 2 STOP
SETH :A1
FD 1
CHECKHEAD
END

TO ABS :VALUE
IF :VALUE <0 OUTPUT - :VALUE
OUTPUT :VALUE
END
```

(a) Run this set of procedures for various values for the variables :A1, :S, and :A2.

(b) Edit the procedure above to start the drawing at any point on the screen.

(c) Use the procedures above to write a procedure for drawing a triangle when given two angles and a not included side of the triangle.

(d) A procedure for drawing a triangle when given two sides and an included angle is not as complicated as several of the others. Write such a procedure.

35. SUMMARY OF COMMANDS

GRAPHIC COMMANDS

To enter Graphics mode type DRAW or execute one of these commands.

FULLSCREEN		In graphics mode, gives full graphics screen.
SPLITSCREEN		Gives mixed text-graphics screen.
FORWARD :distance	FD	Moves turtle forward.
BACK :distance	BK	Moves turtle backwards.
RIGHT :degrees	RT	Rotates turtle.
LEFT :degrees	LT	Rotates turtle.
HOME		Moves turtle to screen center with heading 0.
CLEARSCREEN	CS	Clears graphic screen.
BACKGROUND :color #	BG	Sets screen background to color.
PENCOLOR :color #	PC	Sets the color of the turtle trail.
PENDOWN	PD	Sets turtle to leave trail when moved.
PENUP	PU	Sets turtle to leave no trail.
HIDETURTLE	HT	Makes the turtle disappear.
SHOWTURTLE	ST	Makes the turtle appear.
WRAP		If turtle moves off screen it reappears at opposite edge.
NOWRAP		Undoes the wrap mode.
SETHEADING:degrees	SETH	Rotates turtle to the direction specified.
SETX :x		Horizontally moves turtle to x-coordinate.
SETY :y		Vertically moves turtle to y-coordinate.
SETXY :x :y		Moves turtle to point (x,y).
TOWARDS :x :y		Outputs heading to point (x,y) from turtle.
HEADING		Outputs the turtle's heading in degrees.
XCOR		Outputs the turtle's x-coordinate.
YCOR		Outputs the turtle's y-coordinate.

SPECIAL KEYS

ESC	Deletes last typed character.
Arrow Keys	Move cursor left or right over characters.
CTRL-D	Deletes one character.
CTRL-F	Sets FULLSCREEN in graphics mode.
CTRL-K	Kills rest of line from cursor.
CTRL-S	Sets SPLITSCREEN in graphics mode.
CTRL-T	In graphics mode sets FULLSCREEN to TEXT.

EDITING COMMANDS

Logo screen editor may be entered by typing ED or EDIT, or by typing TO and the name of the procedure you wish to define.

CTRL-A	Moves cursor to beginning of line.
CTRL-B	Moves cursor back 1 screenful of text.
CTRL-C	Saves procedure; exits editor.
CTRL-D	Deletes character under cursor.
CTRL-E	Moves cursor to line's end.
CTRL-G	Exist editor without defining a procedure.

CTRL-K	Deletes all characters from cursor to line's end.
CTRL-L	Scrolls text so line is centered on screen.
CTRL-N	Moves cursor down 1 line.
CTRL-O	Opens a new line at the cursor.
CTRL-P	Moves a cursor up 1 line.

WORKSPACE MANAGEMENT COMMANDS

CATALOG	Lists the files on current disk.
READ "name	Reads a file from disk.
SAVE "name	Saves contents of workspace onto disk.
ERASEFILE :name	Removes a file stored on disk.
READPICT "name	Reads a picture on disk onto screen.
SAVEPICT "name	Saves picture from screen to disk.
ERASEPICT "name	Removes a picture from disk.
PRINTOUT	Prints out the text of procedures currently in workspace. Works with the auxiliary words: ALL NAMES, PROCEDURES.
POTS	Abbreviation for PRINTOUT TITLES.
ERASE name	Erases designated procedures from workspace. Can take qualifiers: ALL, NAMES, PROCEDURES.
GOODBYE	Clears Logo and restarts Logo.